

基于 TPCM 的容器云可信环境研究

刘国杰^{1,2}, 张建标^{1,2}, 杨萍³, 李铮^{1,2}

(1. 北京工业大学信息学部, 北京 100124;

2. 可信计算北京市重点实验室, 北京 100124;

3. 北京信息科技大学, 北京 100192)

摘要: 容器技术是一种轻量级的操作系统虚拟化技术, 被广泛应用于云计算环境, 是云计算领域的研究热点, 其安全性备受关注。提出了一种采用主动免疫可信计算进行容器云可信环境构建方法, 其安全性符合网络安全等级保护标准要求。首先, 通过 TPCM 对容器云服务器进行度量, 由 TPCM 到容器的运行环境建立一条可信链。然后, 通过在 TSB 增加容器可信的度量代理, 实现对容器运行过程的可信度量与可信远程证明。最后, 基于 Docker 与 Kubernetes 建立实验原型并进行实验。实验结果表明, 所提方法能保障云服务器的启动过程与容器运行过程的可信, 符合网络安全等级保护标准测评要求。

关键词: 可信计算; 可信启动; 可信度量; 远程证明

中图分类号: TP309

文献标识码: A

doi: 10.11959/j.issn.2096-109x.2021068

Research on the trusted environment of container cloud based on the TPCM

LIU Guojie^{1,2}, ZHANG Jianbiao^{1,2}, YANG Ping³, LI Zheng^{1,2}

1. Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China

2. Beijing Key Laboratory of Trusted Computing, Beijing 100124, China

3. Beijing Information Science and Technology University, Beijing 100192, China

Abstract: Container technology is a lightweight operating system virtualization technology that is widely used in cloud computing environments and is a research hotspot in the field of cloud computing. The security of container technology has attracted much attention. A method for constructing a trusted environment of container cloud using active immune trusted computing was proposed, and its security meet the requirements of network security level protection standards. First, container cloud servers were measured through the TPCM and a trust chain from the

收稿日期: 2021-01-05; 修回日期: 2021-04-22

通信作者: 张建标, zjb@bjut.edu.cn

基金项目: 国家自然科学基金 (61971014); 国防科技实验信息安全实验室对外开放项目 (2017XXAQ08)

Foundation Items: The National Natural Science Foundation of China (61971014), National Defense Science and Technology Laboratory of Information Security (2017XXAQ08)

论文引用格式: 刘国杰, 张建标, 杨萍, 等. 基于 TPCM 的容器云可信环境研究[J]. 网络与信息安全学报, 2021, 7(4): 164-174.

LIU G J, ZHANG J B, YANG P, et al. Research on the trusted environment of container cloud based on the TPCM[J]. Chinese Journal of Network and Information Security, 2021, 7(4): 164-174.

TPCM to the container's operating environment was established. Then, by adding the trusted measurement agent of the container to the TSB, the trusted measurement and trusted remote attestation of the running process of the container were realized. Finally, an experimental prototype based on Docker and Kubernetes and conduct experiments were built. The experimental results show that the proposed method can ensure the credibility of the boot process of the cloud server and the running process of the container and meet the requirements of the network security level protection standard evaluation.

Keywords: trusted computing, trusted boot, trusted measurement, remote attestation

1 引言

随着信息技术的发展,信息安全问题日益严峻^[1-3]。网络安全等级保护国家标准对中国信息安全工作非常重要。这项标准被应用于网络安全职能部门、网络安全管理部门以及等级测评机构的各项工作。标准要求系统在启动时对信息设备的主板固件、引导程序、操作系统与关键应用程序进行可信验证,并在检测到被恶意篡改后发送报警信号^[4]。

容器技术是一种轻量级的操作系统虚拟化技术。容器技术广泛应用于云计算环境和数据中心的资源管理、系统运维和软件部署,成为云计算领域的业界热点^[5]。与传统的虚拟机相比,容器直接运行在操作系统中,具有更高的性能、更好的可移植性和更快的速度^[6]。在容器环境中,由容器引擎创建并运行容器,常用的容器引擎是 Docker^[7]。基于容器的任务编排与资源调度技术的快速发展,使容器技术越来越受欢迎,常用的任务编排与资源调度应用是 Kubernetes^[8]。

容器广泛应用部署的主要障碍之一是其面临的安全问题。在容器环境中发现了一些安全问题,包括容器应用、容器之间、容器与主机的安全等^[9-11]。容器镜像及其内部程序可能在运行时被主机或外部攻击者利用漏洞恶意篡改。攻击者可能获得对正在运行的容器的访问特权,通过修改服务配置和二进制文件、启动恶意脚本或启动新进程来发起攻击^[12-14]。容器技术提出了确保映像完整性的解决方案,但它们并不覆盖整个服务生命周期,因此存在用户对容器安全性的信任问题。

目前,大多数研究人员使用命名空间隔离和可信计算技术来确保容器的安全运行。文献[15]提出了安全命名空间,使容器能够独立控制其安全性。文献[16]提出了一种轻量级容器云基础设

施的安全监控解决方案,该方案利用远程证明来验证容器云应用程序在其整个生命周期中的软件完整性。文献[17]提出了两种架构解决方案,用于将 vTPM 集成到基于容器的虚拟化模型中,一种方案将 vTPM 模块放到内核中,另一种方案将 vTPM 放到单独的容器中。文献[18]为每个容器创建一个 vTPM,设计了格式统一的签名列表来授权容器可执行文件。文献[19]设计并实现了一个可信增强的 Docker 容器,构造了一条从硬件到容器内部进程和文件的信任链,同时增加了包括进程监控、文件系统度量、网络监控等安全防护模块。文献[20]对传统的信任模型进行了改进,提出了适合容器云环境的信任模型。

可信计算^[21-23]在解决云环境的安全性问题中发挥了重要作用。TCG (Trusted Computing Group) 提出的解决方案是以 TPM (trusted platform module) 为可信根,作为外挂部件被动调用无法进行主动度量^[24]。我国学者提出了 TPCM (trusted platform control module)^[25]作为可信计算环境的信任根,它是可信计算的核心模块。TPCM 与 TSB (trusted software base)、TCA (trust connection architecture) 等功能模块,实现主动免疫可信计算。

本文提出了一种采用主动免疫可信计算构建容器云可信环境的方法,主要贡献如下:① 提出了基于主动免疫可信计算的容器云可信环境架构,可以全面保护容器云环境;② 提出了利用 TPCM 进行可信启动的流程,从 TPCM 到容器运行环境与容器运行进程建立一条可信链;③ 基于容器的命名空间隔离机制,增加了 TSB 对容器应用的度量功能,实现容器的主动度量。

2 研究背景

为了研究容器云可信环境的构建方法,本节

主要进行容器云架构、主动免疫可信计算等相关研究工作。容器云架构是近年来出现并发展迅速的新技术,针对容器云架构的可信计算技术需要进一步深入研究。主动免疫可信计算技术是我国近年来快速发展的安全防护技术。本文将其应用于容器云环境的安全防护。

2.1 可信平台控制模块

中关村可信计算联盟组织提出了 TPCM, 中国国家标准《信息安全技术可信计算规范 可信平台控制模块》被制定, 并发布了征求意见稿。这个标准阐明了 TPCM 的功能组成。TPCM 主要包括嵌入了可信密码模块核心、基础软件、功能服务、功能接口和计算部件接口的 CPU。图 1 显示了 TPCM 的功能结构^[26]。

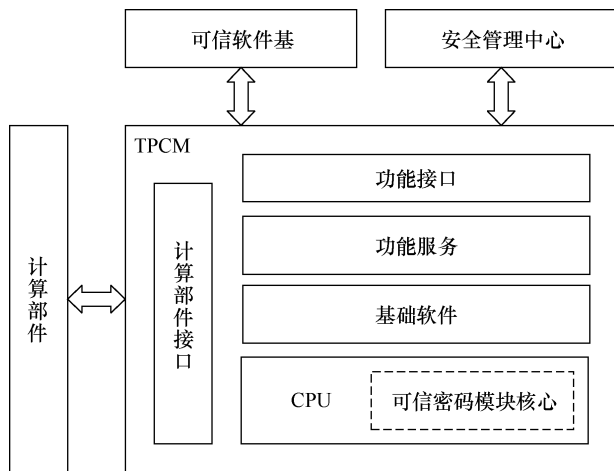


图 1 TPCM 的功能结构
Figure 1 The functional structure of TPCM

TPCM 在 TCM (trusted cryptography module) 的基础上增加信任根控制功能, 实现密码与控制相结合, 将可信平台控制模块设计为可信计算控制节点, 实现了 TPCM 对整个平台的主动控制; 在可信平台主板中增加了可信度量控制节点, 实现了计算和可信双节点融合; 软件基础层实现宿主操作系统和可信软件基的双重系统核心, 通过在操作系统核心层并接一个可信的控制软件接管系统调用, 在不改变应用软件的前提下实施对应执行点的可信验证, 达到主动防御效果^[27]。

2.2 可信软件基

TSB 由基本信任基、主动监控机制、可信基准库、支撑机制与协作机制组成^[28]。图 2 展示了

TSB 的组成结构。基本信任基在启动过程中实现对其他机制的验证与加载。主动监控机制包括控制机制、度量机制与判定机制, 拦截应用的系统调用, 实现对相关的主体、客体、操作与环境的度量与控制。TSB 通过支撑机制实现对可信根实体的资源访问。TSB 通过协作机制与可信策略管理中心交互。本文扩展了 TSB 的功能实现对容器的主动可信度量与控制。

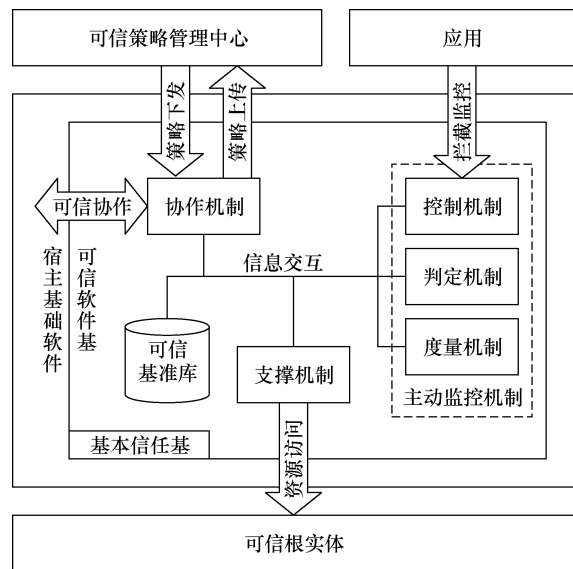


图 2 TSB 组成结构
Figure 2 Composition structure of TSB

2.3 主动免疫可信计算

主动免疫可信计算的核心是构建基于 TPCM 的双系统体系结构, 如图 3 所示。TPCM、TSB 可信的度量和主动监控系统启动环境和运行环境, 实现计算节点的主动免疫可信计算。TPCM 作为可信根, 先于 CPU 通电, 对主板硬件进行主动度量。TSB 承接底层硬件的信任链, 并利用 TPCM 对系统运行环境进行主动度量, 对系统运行环境进行主动监控。从系统启动开始, 以 TPCM 为可信根, 通过信任链传递机制, 逐级实施主动度量验证。通过 TCA 保证计算节点的网络环境的可信^[29]。

2.4 容器云架构

容器云架构主要包括主节点与计算节点两部分, 如图 4 所示。容器云架构的主节点一般采用 Kubernetes 作为管理程序, 其主要部件运行在容器中, 包括 API Server、Controller、Scheduler、Etc 等关键部件。容器云架构的计算节点一般采

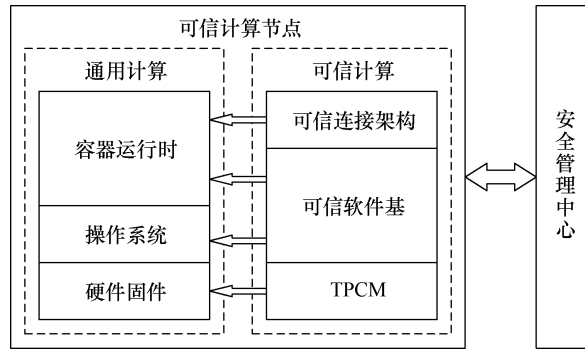


图3 主动免疫可信计算节点
Figure 3 Active immune trusted computing node

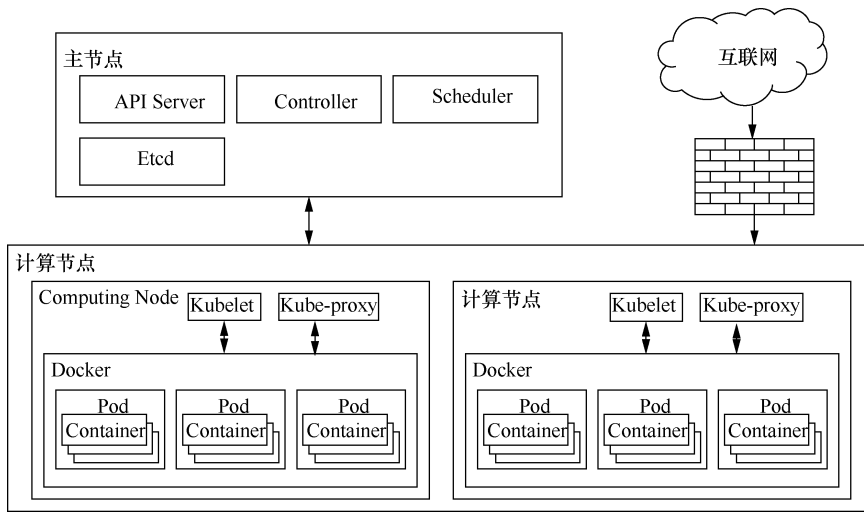


图4 容器云架构
Figure 4 Container-based cloud architecture

用Docker作为容器应用的运行程序,包括Kubelet与Kube-proxy等关键部件实现与Kubernetes交互和负载均衡。

3 相关研究

3.1 基于 TPCM 的可信启动

TCG提出的解决方案,以TPM为信任根作为外挂部件被动调用,无法进行主动度量^[30]。本文提出了一种基于TPCM的服务器PXE可信启动方法^[31],图5展示了PXE可信启动过程。



图5 PXE可信启动过程
Figure 5 PXE trusted boot process

3.2 vTPM 保护容器

文献[32]提出了虚拟化TPM的软件实现,

用于保护在Hypervisor运行的多个虚拟机。每个虚拟机对应一个vTPM。这种vTPM设计如图6(a)所示。文献[17]提出了两种架构解决方案,用于将vTPM集成到基于容器的虚拟化模型中,一种方案将vTPM模块放到内核中,另一种方案将vTPM放到单独的容器中。图6(b)展示了vTPM放到内核中的功能设计。图6(c)展示了在一个特权容器中运行vTPMManager,并且在每个容器中运行vTPMAdapter。

3.3 容器完整性认证

文献[16]提出了一种轻量级的云环境下Docker容器的完整性验证方案DIVE。图7展示了DIVE远程证明的工作流程。DIVE利用远程认证来验证云应用程序在其整个生命周期中的软件完整性。DIVE不能防护内存攻击,需要额外保存内存。

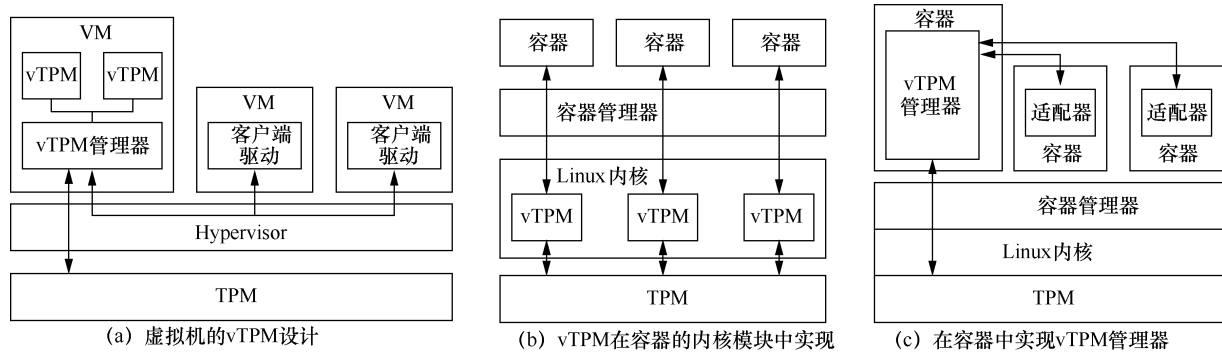


图6 3种不同的vTPM设计
Figure 6 Three different vTPM designs

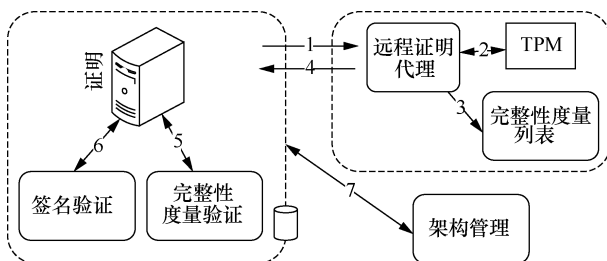


图7 DIVE 远程证明的工作流程
Figure 7 Work flow of cave remote certification

4 容器云可信架构

本文基于主动免疫可信计算技术来构建容器云可信环境，其架构如图8所示。容器云可信环境架构包括安全管理中心、部署在主节点

与计算节点的TPCM与TSB。安全管理中心主要包括系统管理、安全管理与审计管理3个功能模块，主要进行系统管理、安全策略的制定与下发、容器应用白名单的维护、系统运行信息的审计等。

4.1 可信启动

容器云可信的启动过程如图9所示。基于TPCM构建的可信主板在启动时，TPCM先于主板上电。TPCM度量主板固件和BIOS，如果没问题进行下一步骤，否则发出报警信号。然后主板上电，BIOS获得控制权。BIOS度量操作系统内核与TSB，如果没问题进行下一步骤，否则发出报警信号。操作系统内核启动，TSB获得控制权。

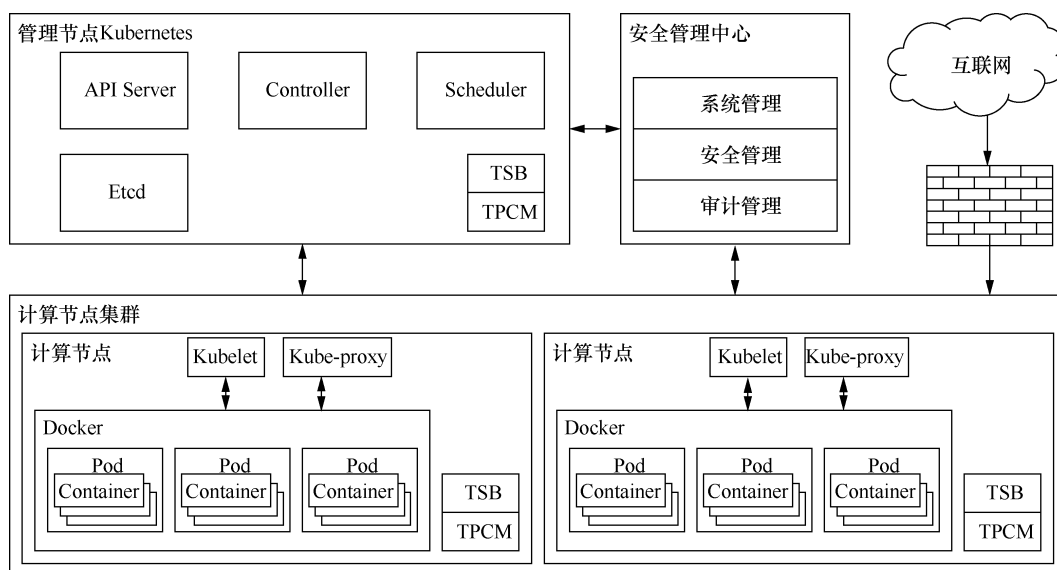


图8 容器云可信环境架构
Figure 8 Trusted container cloud environment architecture

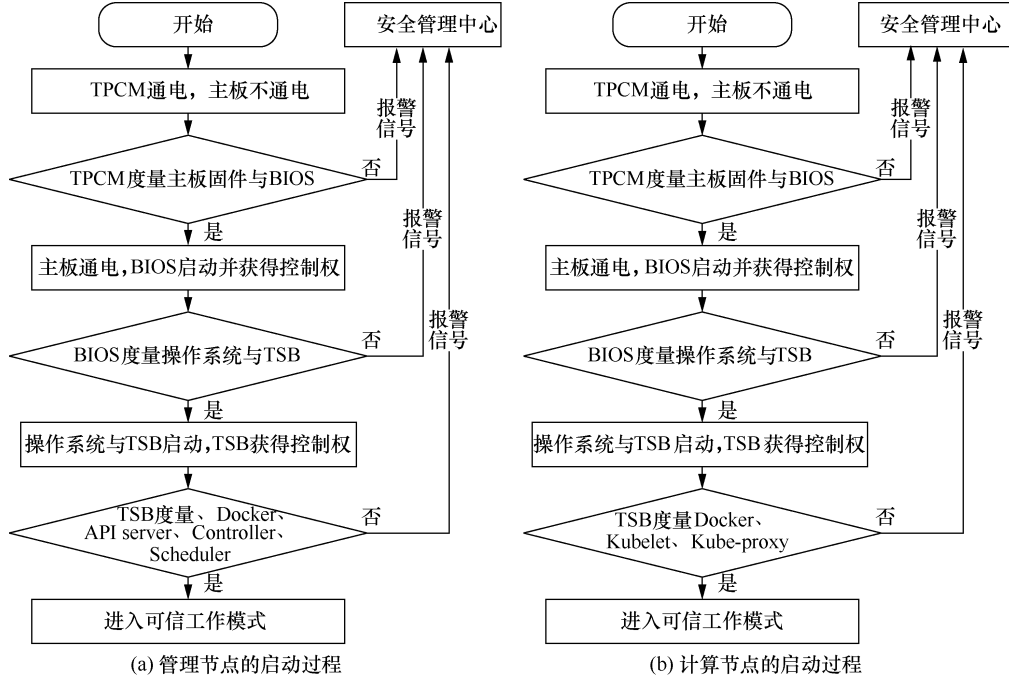


图9 容器云可信的启动过程
Figure 9 Container based cloud trusted startup process

在容器云环境的主节点, TSB 度量 Docker、API Server、Controller、Scheduler 等关键应用程序与配置文件。在容器云环境的计算节点, TSB 度量 Docker、Kubelet、Kube-proxy 等关键应用程序与配置文件。

定义 1 每个启动阶段的一个关键部件作为一个实体 E 。

定义 2 第一个启动阶段的关键部件作为实体 E_0 , 称为信任根。

定义 3 在启动过程中, 上一个实体 E_{i-1} 启动当前实体 E_i , 其中 $1 \leq i \leq n$ 。如果每个启动阶段的每一个关键实体都验证为可信, 则整个启动过程是可信的。为了确保当前实体 E_i 的可信性, 必须首先信任上一个实体 E_{i-1} , 该信任模型为

$$E_{i-1} \xrightarrow{v_{i-1}} E_i (1 \leq i \leq n) \quad (1)$$

式(1)中, v_{i-1} 是上一个实体 E_{i-1} 的可信验证模块, 用于对当前实体的可信性验证。

以信任根为起点, 对实体进行逐级度量与验证, 逐级信任, 得到一个信任链。

$$E_0 \rightarrow E_1 \rightarrow E_2 \rightarrow \cdots \rightarrow E_n \quad (2)$$

本文提出的可信启动模型的实体为 TPCM、

BIOS、TSB、Master 节点与 Computing 节点的关键应用, 分别用实体 E_{TPCM} 、 E_{BIOS} 、 E_{TSB} 、 E_{Master} 与 $E_{\text{Computing}}$ 表示, 以 E_{TPCM} 为信任根得到可信启动的验证模型为

$$E_{\text{TPCM}} \xrightarrow{v_{\text{TPCM}}} E_{\text{BIOS}} \xrightarrow{v_{\text{BIOS}}} E_{\text{TSB}} \xrightarrow{v_{\text{TSB}}} E_{\text{Master}} \quad (3)$$

$$E_{\text{TPCM}} \xrightarrow{v_{\text{TPCM}}} E_{\text{BIOS}} \xrightarrow{v_{\text{BIOS}}} E_{\text{TSB}} \xrightarrow{v_{\text{TSB}}} E_{\text{Computing}} \quad (4)$$

式中 v_{TPCM} 、 v_{BIOS} 、 v_{TSB} 为可信度量与验证模块, 用于上一个实体对当前实体的可信度量与验证。

以 TPCM 作为起点, 进行逐级度量与验证, 最终得到 Master 节点与 Computing 节点的信任链。

$$E_{\text{TPCM}} \rightarrow E_{\text{BIOS}} \rightarrow E_{\text{TSB}} \rightarrow E_{\text{Master}} \quad (5)$$

$$E_{\text{TPCM}} \rightarrow E_{\text{BIOS}} \rightarrow E_{\text{TSB}} \rightarrow E_{\text{Computing}} \quad (6)$$

在容器云的控制节点与计算节点启动时、运行环境的信任链建立后, TPCM 通过容器引擎与 TSB 实现对容器应用文件与进程的度量。容器的运行与虚拟机的运行有很多不同, 同一台宿主机的容器共享操作系统内核, 而虚拟机必须独立安装运行操作系统。

4.2 主动可信度量

主动可信度量把 TPCM 作为主动的度量根,

其检查原则是 TPCM 先检查硬件设备配置, 再检查软件系统。对硬件设备先检查身份合法性, 再检查固件完整性。对软件系统先检查输入数据的真实性和新鲜性, 再检查输入数据的完整性, 最后通过密码机制保护信任链, 构建可信计算环境^[33]。

可信策略管理中心为每一个容器 Image 制定可信策略。TSB 根据容器可信度量代理下载的容器, 从可信策略管理中心下载该容器的可信策略。容器云环境的主动可信度量如图 10 所示。TPCM 为 TSB 的可信功能提供支撑。TSB 根据可信策略通过嵌入容器引擎中的可信度量代理对容器应用进行主动可信度量, 可信度量代理对容器应用度量后, 把度量结果返回到 TSB。

在容器启动时, 会生成唯一的命名空间, 并把容器镜像加载到内存中, 这时, 可信度量代理检测到容器的加载, 建立对应的可信度量缓存。根据容器 ImageID 找到对应的可信策略并加载到内存中。完整性度量列表数据结构如图 11 所示。

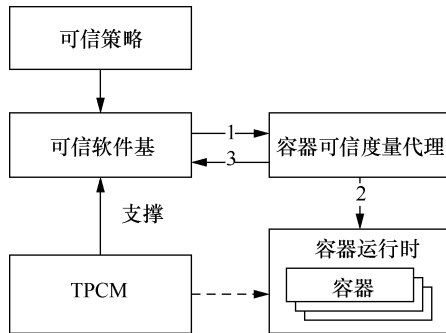


图 10 容器可信度量流程
Figure 10 Container trusted measurement process

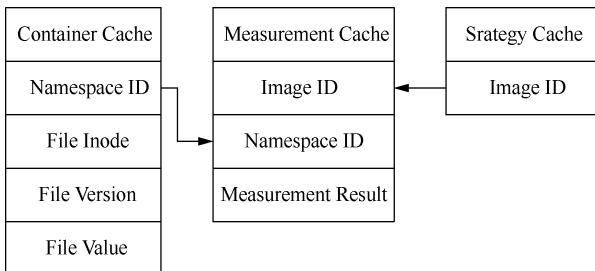


图 11 完整性度量列表数据结构
Figure 11 Data structure of integrity measurement list

容器的可信度量流程: 首先, 根据容器加载建立对应的缓存; 然后, 根据容器 ImageID 查找对应的可信策略; 最后, 根据可信策略对容器缓存进行度量, 得出度量结果, 存储到度量缓存中。

4.3 可信远程证明

本文提出了一种容器云环境的可信远程证明架构, 该架构可以从主节点与计算节点收集和验证可信远程证明信息, 并在安全管理中心集中管理这些信息。该架构通过验证主节点与计算节点的受信任状态, 来验证容器云环境的受信任证明。可信远程证明流程如图 12 所示。终端用户发起可信远程证明请求到达安全管理中心, 安全管理中心通过验证容器云环境的主节点与计算节点的受信任状态, 完成容器云环境的受信任证明。安全管理中心把远程证明结果返回给终端用户。

以下是可信远程证明的具体步骤。

步骤 1 挑战者 A 应请求者 B 的请求, 创建一个不可预知的随机数 N 发给证明节点的证明代理 T。

步骤 2 证明代理 T 以 Quote 请求的形式发送给计算节点 C 的 TPCM。

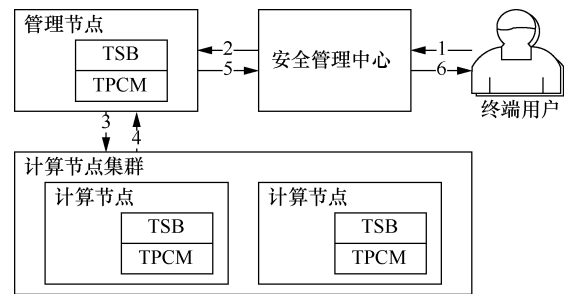


图 12 可信远程证明过程
Figure 12 Trusted remote attestation process

步骤 3 计算节点 C 的 TPCM 使用存储根密钥 SRK, 读取身份证明密钥 AIK, 执行 Quote 命令。

步骤 4 证明代理 T 签名 Quote, 并把签名后的 Quote、完整性状态 S、身份证明密钥 (AIK) 证书与度量存储日志 L 发送给挑战者 A。

步骤 5 挑战者 A 验证身份证明密钥证书的有效性, 同时验证 Quote 签名的正确性和新鲜性。

步骤 6 挑战者 A 将度量存储日志 Log 与基准值 BV 比较, 如果度量值与基准值匹配, 证明远程节点处于可信状态。

5 实验与原型

5.1 实验设置

(1) 可信启动需要在服务器主板增加 TPCM

芯片, 修改并配置服务器的通电顺序。当计算机通电时, CPLD 控制逻辑首先为 TPCM、闪存与高速开关通电, 其他设备不通电, 通过可信根 TPCM 验证启动代码与主板固件的可信性; 然后在 TPCM 的控制下为其他设备通电与开机^[34-35]。

(2) 可信度量与可信远程证明基于软件 TPCM 仿真器在虚拟机上完成。实验中的开发环境的配置是: 8 Core CPU 3.1 GHz, 32 GB Memory, Ubuntu 18.04.5, Docker CE 19.03.13, Golang 1.13.15。实验中的测试环境需要使用 VMware 虚拟化软件创建 3 台虚拟机, 这 3 台虚拟机的网络拓扑结构与详细配置如图 13 所示。主节点的配置为: 2 Core CPU 3.1 GHz, 4 GB Memory, Ubuntu 18.04.5, Docker CE 19.03.13, Kubernetes 1.18.8, Attestation Server and Software TPCM Emulator。计算节点的配置为: 2 Core CPU 3.1 GHz, 4 GB Memory, Ubuntu 18.04.5, Docker CE 19.03.13, Attestation Client and Software TPCM Emulator。Attestation Client、Attestation Server and Software TPCM Emulator 基于 IBM TPM 2.0 Emulator^[36]开发。

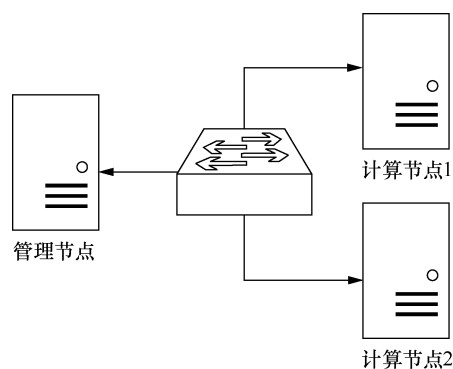


图 13 实验环境的网络拓扑结构
Figure 13 Network topology of experimental environment

为了实现提出的方法, 需要配置 TPCM 并修改 BIOS 固件的源代码, 增加对 OS 引导程序的可信验证模块 (BIOS 固件使用了开源的 EDK2 源代码^[37]); 需要把编译生成的 FV 文件通过专用工具烧写至 BIOS 闪存。容器的 runtime 使用开源的 Docker 源代码^[38]。容器的管理调度使用开源的 Kubernetes 源代码^[39]。

5.2 实验与分析

本文从以下 3 方面对实验结果进行分析。

(1) 可信启动。首先, TPCM 对启动过程中的启动代码、主板固件、系统内核、TSB、Docker、Kubernetes 等关键部件进行度量并存储基准值。然后, 在运行启动过程中分别对这些部件进行可信度量并与基准值比较。实验结果表明, TPCM 能够发现被修改的固件并发出报警信号。本文所采用的 TPCM 与文献[16,19]采用 TPM 的关键功能的对比, 如表 1 所示。

本文进行了可信启动的性能测试, 在操作系统 Ubuntu16.04.7 上, 经过 20 次的可信启动实验, 结果表明对操作系统进行可信度量时间比重为 0.55% 左右。这个结果是可以接受的, 对操作系统时间的启动影响不大。

(2) 主动可信度量。本文基于 TPCM 在 Docker 中嵌入可信度量代理, 在容器启动时对容器进行度量, 如果与基准值一样, 则正常运行, 否则发出报警信号并终止运行。本文测试了容器被恶意篡改前后的运行结果, 只有没有被篡改的容器能够运行。

把 TSB 与容器的可信度量代理部署到管理节点与计算节点进行实验。在相同条件下, 使用文献[17-18]中的 vTPM 进行实验并与本文方法进行比较。本文方法不需要额外加载 vTPM, 节约了时间并提高了效率, 结果如图 14 所示。

表 1 TPCM 与 TPM 防御有效性的比较
Table 1 Comparison of defense effectiveness between TPCM and TPM

攻击类型	TPCM 的防御效果	TPM 的防御效果
修改 MBR 启动代码	实时发现并阻止操作系统加载	TPM 不能发现这个问题, 不能阻止启动
BIOS Bootblock 入侵	实时发现并阻止启动	无此功能
Main BIOS 入侵	实时发现并阻止启动	TPM 不能发现这个问题, 不能阻止启动
使用未经认证的 PCIE 设备	实时发现并阻止启动	TPM 不能发现这个问题, 不能阻止启动
破坏平台完整性	实时发现并阻止启动	TPM 不能发现这个问题, 不能阻止启动

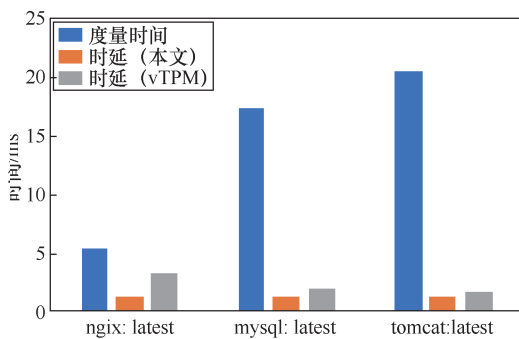


图 14 容器 Image 的度量时间
图 14 Measurement time of container Image

(3) 可信远程证明。可信远程证明包括对容器运行环境的证明与容器自身的证明。实验表明, 本文方法能够根据用户的请求得到容器运行环境的可信状态数据。

5.3 与相关工作比较

本文采用主动免疫可信计算技术来保护容器云环境的安全可信运行。本文对比了基于 TPM^[16,19]或基于 vTPM^[17-18]构建基于容器的可信云环境方法, 总结如表 2 所示。

在可信启动方面, 由于 TPCM 先上电, 在主板通电之前能够对服务器主板固件与 BIOS 进行可信验证。根据表 1 中的 TPCM 与 TPM 防御效果比较得出, TPCM 安全性比文献[16,19]提出的采用 TPM 进行可信启动安全性高, 本文提出的可信启动方法能够验证启动代码与主板固件。

在可信度量方面, 本文基于 TPCM 采用在 Docker 中嵌入可信度量代理进行主动度量, 实际由可信芯片 TPCM 进行可信度量。TPCM 物理芯片不能被恶意篡改, 而虚拟的 vTPM 可能被恶意篡改。根据表 2 得出, 本文提出的使用 TPCM 来保护容器的方法的安全性比文献[17-18]提出的使

用 vTPM 来保护容器的方法好。实验结果表明, 本文提出的方法不需要加载 vTPM, 节约了初始化时间, 提高了效率。

在可信远程证明方面, 本文方法能够验证启动代码与主板固件, 并且实现了基于 TPCM 的主动度量, 能够实时证明容器云环境的安全可信。

6 结束语

本文方法实现了容器云环境的可信启动、主动可信度量与可信远程证明, 实现了容器云可信环境。首先, 基于 TPCM 实现可信启动。从 TPCM 到 BIOS, 再到系统内核与 TSB, 最后到 Docker 与 Kubernetes 应用程序, 逐级进行可信度量, 形成一条信任链。其次, 将可信度量代理模块嵌入 Docker, 实现对通过 Docker 中运行的容器进行主动度量, 这比采用 TPM 或 vTPM 保护容器的方法, 安全性更高并且执行效率更高。最后, 本文实现了对容器云环境的可信远程证明。本文采用了 TPCM, 可以保护硬件固件的安全性、容器所在主机的操作系统内核、Docker 与 Kubernetes 的安全性, 并且通过可信度量代理对运行的容器进行主动度量, 保护范围更全面且可信性更高。

下一步, 将在安装有物理芯片 TPCM 的容器云服务上对本文提出的方法进行验证。由于实验条件限制, 本文采用虚拟机对所提方法进行验证, 没有使用 TPCM, 虽然可以验证所提方法的安全性, 但仍需要在真实环境下进行更多的实验。

参考文献:

- [1] 张玉清, 王晓菲, 刘雪峰, 等. 云计算环境安全综述[J]. 软件学

表 2 构建容器的不同受信任的云环境方法的比较
Table 2 Comparison of different trusted cloud environment methods for building containers

功能类别	TPCM	TPM 或 vTPM
是否能够保护 BIOS 固件	先于主 CPU 启动, 能够验证 BIOS 固件	TPM 与 vTPM 是被动调用, 不能验证 BIOS 固件
可信度量时间	只需要初始化一次, 相对 vTPM 较短	vTPM 需要每次启动容器是初始化, 时间较长
是否易受攻击	采用物理芯片, 作为信任根, 不易受攻击	vTPM 容易被恶意篡改
是否需要外部保护	采用物理芯片, 作为信任根, 不需要额外保护	vTPM 是虚拟的, 作为 OS 的一个进程, 容易被破坏篡改, 需要保护
远程证明复杂度	采用可信度量代理, 每个容器根据度量策略, 生成一个度量列表, 对于同一台主机调用一次 TPCM 就可以完成验证与证明	每个容器对应一个 vTPM, 对于同一台主机的多个容器需要调用每一个容器的 vTPM 进行验证与证明

- 报, 2016, 27(6): 1328-1348.
- ZHANG Y Q, WANG X F, LIU X F, et al. Survey on cloud computing security[J]. Journal of Software, 2016, 27(6): 1328-1348.
- [2] SINGH A, CHATTERJEE K. Cloud security issues and challenges: a survey[J]. Journal of Network and Computer Applications, 2017, 79.
- [3] SINGH S, JEONG Y S, PARK J H. A survey on cloud computing security: issues, threats, and solutions[J]. Journal of Network and Computer Applications, 2016, 75.
- [4] 陈广勇, 祝国邦, 范春玲. 《信息安全技术网络安全等级保护测评要求》(GB/T 28448-2019) 标准解读[J]. 信息安全杂志, 2019, 19(7): 1-8.
- CHEN G Y, ZHU G B, FAN C L. Information security technology-evaluation requirement for classified protection of cybersecurity (GB/T 28448-2019) standard interpretation[J]. Netinfo Security, 2019, 19(7): 1-8.
- [5] 吴松, 王坤, 金海. 操作系统虚拟化的研究现状与展望[J]. 计算机研究与发展, 2019, 56(1): 58-68.
- WU S, WANG K, JIN H. Research situation and prospects of operating system virtualization[J]. Journal of Computer Research and Development, 2019, 56(1): 58-68.
- [6] FELTER W, FERREIRA A, RAJAMONY R, et al. An updated performance comparison of virtual machines and Linux containers[C]//2015 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS). 2015.
- [7] MERKEL D. Docker: lightweight linux containers for consistent development and deployment[J]. Linux Journal, 2014(1):76-90.
- [8] BRENDAN B, BRIAN G, DAVID O, et al. Borg, Omega, and Kubernetes[J]. Communications of the ACM, 2016, 59(5):50-57.
- [9] SULTAN S, AHMAD I, DIMITRIOU T. Container security: issues, challenges, and the road ahead[J]. IEEE Access, 2019, 7(1): 52976-52996.
- [10] RUI S, GU X H, ENCK W. A study of security vulnerabilities on docker hub[C]//Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy. 2017.
- [11] THANH B. Analysis of docker security[J]. arXiv preprint arXiv:1501.02967, 2015.
- [12] 鲁涛, 陈杰, 史军. Docker 安全性研究[J]. 计算机技术与发展, 2018, 28(6): 115-120.
- LU T, CHEN J, SHI J. Research of Docker security[J]. Computer Technology and Development, 2018, 28(6): 115-120.
- [13] JAYANTH G, DESIKAN T, TURNER Y. Over 30% of official images in docker hub contain high priority security vulnerabilities[R]. Technical Report. 2015.
- [14] MANU A R, PATEL J K, AKHTAR S, et al. A study, analysis and deep dive on cloud PAAS security in terms of Docker container security[C]//2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT). 2016.
- [15] SUN Y Q, SAFFORD D, ZOHAR M, et al. Security namespace: making Linux security frameworks available to containers[C]//Proceedings of the 27th USENIX Conference on Security Symposium (SEC'18). 2018: 1423-1439.
- [16] BENEDICTIS M D, LIOY A. Integrity verification of Docker containers for a lightweight cloud environment[J]. Future Generation Computer Systems, 2019, 97(8):236-246.
- [17] HOSSEINZADEH S, LAURÉN S, LEPPNEN V. Security in container-based virtualization through vTPM[C]//IEEE/ACM International Conference on Utility & Cloud Computing. 2017.
- [18] GUO Y, YU A, GONG X, et al. Building trust in container environment[C]//2019 18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/13th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE). 2019.
- [19] 王鹏, 胡威, 张雨菡, 等. 基于 Docker 的可信容器[J]. 武汉大学学报(理学版), 2017, 63(2): 102-108.
- WANG J, HU W, ZHANG Y H, et al. Trusted container based on Docker[J]. Journal of Wuhan University (Natural Science Edition), 2017, 63(2): 102-108.
- [20] XIE X L, YUAN T W, ZHOU X, et al. Research on trust model in container-based cloud service[J]. CMC: Computers, Materials & Continua, 2018, 56(2).
- [21] 冯登国, 秦宇, 汪丹, 等. 可信计算技术研究[J]. 计算机研究与发展, 2011, 48(8): 1332-1349.
- FENG D G, QIN Y, WANG D, et al. Research on trusted computing technology[J]. Journal of Computer Research and Development, 2011, 48(8): 1332-1349.
- [22] SHEN C X, ZHANG H G, WANG H M, et al. Research on trusted computing and its development[J]. Science China Information Sciences, 2010, 53(3): 405-433.
- [23] 张焕国, 罗捷, 金刚, 等. 可信计算研究进展[J]. 武汉大学学报(理学版), 2006(5): 513-518.
- ZHANG H G, LUO J, JIN G, et al. Development of trusted computing research[J]. Wuhan University Journal of Natural Sciences, 2006(5): 513-518.
- [24] FENG D G, QIN Y, FENG W, et al. The theory and practice in the evolution of trusted computing[J]. Chinese Science Bulletin, 2014, 59(32):4173-4189.
- [25] 郭颖, 毛军捷, 张翀斌, 等. 基于可信平台控制模块的主动度量方法[J]. 清华大学学报(自然科学版), 2012, 52(10): 1465-1473.
- GUO Y, MAO J J, ZHANG C B, et al. Active measures base on a trusted platform control module[J]. Journal of Tsinghua University (Sci and Tech), 2012, 52(10): 1465-1473.
- [26] 中国国家标准委员会. 信息安全技术可信计算规范可信平台控制模块[S]. 征求意见稿, 2019.
- National Information Security Standardization Technical Committee. Information security technology-trusted computing specification-motherboard function and interface of trusted platform[S]. Draft for Comments, 2019.
- [27] 沈昌祥. 用主动免疫可信计算 3.0 筑牢网络安全防线营造清朗的网络空间[J]. 信息安全研究, 2018, 4(4): 282-302.
- SHEN C X. To create a positive cyberspace by safeguarding network security with active immune trusted computing 3.0[J]. Journal of Information Security Research, 2018, 4(14): 282-302.
- [28] GB/T 37935-2019. 信息安全技术 可信计算规范 可信软件基[S]. 2020.
- GB/T 37935-2019, Information security technology-Trust computing specification-Trust software base[S]. 2020.
- [29] 沈昌祥, 张大伟, 刘吉强, 等. 可信 3.0 战略:可信计算的革命性演变[J]. 中国工程科学, 2016, 18(6): 53-57.

SHEN C X, ZHANG D W, LIU J Q, et al. The strategy of TC 3.0: a revolutionary evolution in trusted computing[J]. Strategic Study of Chinese Academy of Engineering, 2016, 18(6): 53-57.

- [30] 田俊峰, 常方舒. 基于 TPM 联盟的可信云平台管理模型[J]. 通信学报, 2016, 37(2): 1-10.

TIAN J F, CHANG F S. Trusted cloud platform management model based on TPM alliance[J]. Journal on Communications, 2016, 37(2): 1-10.

- [31] 刘国杰, 张建标. 基于 TPCM 的服务器可信 PXE 启动方法[J]. 网络与信息安全学报, 2020, 6(6): 105-111.

LIU G J, ZHANG J B. TPCM-based trusted PXE boot method for servers[J]. Chinese Journal of Network and Information Security, 2020, 6(6): 105-111.

- [32] BERGER S, CÁCERES R, GOLDMAN K A, et al. vTPM: virtualizing the trusted platform module[J]. Usenix Security, 2006, 15: 305-320.

- [33] 田健生, 詹静. 基于 TPCM 的主动动态度量机制的研究与实现[J]. 信息网络安全, 2016, 16(6): 22-27.

TIAN J S, ZHAN J. Research and implementation of active dynamic measurement based on TPCM[J]. Netinfo Security, 2016, 16(6): 22-27.

- [34] 黄坚会, 石文昌. 基于 ATX 主板的 TPCM 主动度量及电源控制设计[J]. 信息网络安全, 2016, 16(11): 1-5.

HUANG J H, SHI W C. The TPCM active measurement and power control design for ATX motherboard[J]. Netinfo Security, 2016, 16(11): 1-5.

- [35] 黄坚会, 沈昌祥. TPCM 主动防御可信服务器平台设计[J]. 郑州大学学报(理学版), 2019, 51(3): 1-6.

HUANG J H, SHEN C X. Trusted platform design of server with TPCM active defense[J]. Journal of Zhengzhou University (Natural Science Edition) (2019), 51(3): 1-6.

- [36] IBM's software TPM 2 [EB].

- [37] EDK II project [EB].

- [38] Docker [EB].

- [39] Kubernetes [EB].

[作者简介]



刘国杰 (1982-), 男, 北京工业大学博士生, 主要研究方向为网络与信息安全、可信计算。



张建标 (1969-), 男, 北京工业大学教授、博士生导师, 主要研究方向为网络与信息安全、可信计算。



杨萍 (1987-), 女, 北京信息科技大学讲师, 主要研究方向为人工智能、智能信息处理、机器学习和信息安全。



李铮 (1992-), 女, 北京工业大学讲师, 主要研究方向为信息安全、密码分析和对称密码算法的设计。